Efficient Topology Construction for RPL over IEEE 802.15.4 in Wireless Sensor Networks

Bogdan Pavkovic[a] , Andrzej Duda[a] Won-Joo Hwang[b]
Fabrice Theoleyre[c,*]

[a]*Grenoble Institute of Technology, CNRS Grenoble Informatics Laboratory, UMR 5217, F-38400 Saint Martin d'Heres, France*

[b]*Inje University, 607 Obang-dong, 621-749 Gimhae, Korea*

[c]*CNRS - ICUBE, University of Strasbourg, UMR 7357, Boulevard Sebastien Brant, F-67412 Illkirch Cedex, France*

Abstract

IEEE 802.15.4-2006 represents a widely used standard for multihop Wireless Sensor Networks. However, the standard exploits a tree structure in the MAC layer, which may lead to network partitions even after a single link or node failure, i.e. the well known single point of failure problem. Besides, the single path approach avoids the routing protocol to select by itself a next hop based on its own criteria. Moreover, transmissions must be appropriately scheduled in the IEEE 802.15.4 cluster-tree to avoid collisions. In this paper, we propose to modify the cluster-tree structure into a Cluster-Directed Acyclic Graph (DAG) to improve the robustness and the topology redundancy at the MAC layer. We also present a simple greedy scheduling algorithm integrated with the IEEE 802.15.4 MAC mechanisms. Simulations show that the proposed mechanisms optimize the MAC layer for multihop topologies. In particular, the routing protocol (e.g. RPL) is able to exploit efficiently the cluster-DAG and to reduce the number of packet losses and the end-to-end delay. Last but not least, the cluster-DAG structure leads globally to energy savings by reducing the number of transmissions at the MAC layer.

Key words: IEEE 802.15.4; cluster-tree; Directed Acyclic Graph; multipath; redundancy

---

[*] Corresponding author.

Email addresses: `bogdan.pavkovic@imag.fr` (Bogdan Pavkovic), `andrzej.duda@imag.fr` (Andrzej Duda), `ichwang@inje.ac.kr` (Won-Joo Hwang), `theoleyre@unistra.fr` (Fabrice Theoleyre).

[1] This work was partially supported by the French National Research Agency (ANR) project IRIS under contract ANR-11-INFR-016 and by the Korean and

Preprint submitted to Ad Hoc Networks 29 November 2013

1 Introduction

The IEEE 802.15.4-2006 standard specifies the PHY and MAC layers for Low Rate Wireless PAN (LR-WPAN) [1]. The standard aims at enabling low-cost communications for Wireless Sensor and Actuator Networks. Among other applications, it may be used in smart home environments for energy management, air conditioning and security systems [15].

IEEE 802.15.4 was initially designed mainly for single hop networks: the PAN coordinator serves as a gateway to the Internet and is directly connected to the end-devices. This *star* topology is particularly efficient to save energy: only the PAN coordinator has to stay awake, and end-devices may turn off their radio if they don't have packets to transmit.

In multihop, the *beacon-enabled mode* of IEEE 802.15.4 enables to save energy by adopting a superframe structure. During the active part of the superframe, a node has to stay awake to receive and transmit packets. Then, it switches off its radio during the inactive part.

To limit the number of collisions, the active parts of interfering nodes should not overlap. Thus, IEEE 802.15.4-2006 proposes to create a tree structure (cluster-tree), rooted at the PAN coordinator. All the nodes which accept to forward packets from their children are designated as *coordinators* and constitute the non-leaf nodes in the cluster-tree. The cluster-tree structure helps to schedule accurately the active parts for each coordinator: a child stays awake during the active part of its parent to transmit packets while it has its own non overlapping active part for its own children.

However, the cluster-tree structure is prone to failures since a node forwards its whole traffic to a selected coordinator. In the case of a link or a node failure, network disconnection times can be potentially large and lead to extensive packet drops due to limited buffer sizes. Moreover, recent proposals advocate the use of multiple paths in wireless sensor networks to create stable routes [5] and to improve security [26]. Clearly, the IEEE 802.15.4 cluster-tree structure cannot offer such features: only one single path toward the PAN coordinator is available.

Because of the tree organization, beacons may be scheduled simultaneously. The Beacon Only Period (BOP) proposed to schedule the beacons for interfering coordinators to avoid collisions [19]. Because data packets may also collide, Villaverde et al. have proposed to schedule appropriately the active parts of the superframes for all the coordinators [28].

French Foreign Ministers under project PHC/STAR DCS.

The IETF ROLL working group proposed the IPv6 Routing Protocol for Low power and Lossy Networks (RPL) [30]. RPL is a Distance Vector routing protocol that builds a DODAG (Destination Oriented Directed Acyclic Graph) anchored at a border router (PAN coordinator) of a WSN. A node maintains several parents to construct different routes toward the sink.

The goal of this paper is to investigate the possibility to enable the joint operation of two emerging standards — RPL and IEEE 802.15.4. Our aim is to create a MAC topology compliant with the RPL requirements: several parents may be chosen to offer a minimum redundancy and diversity. Besides, we aim also at limiting the number of collisions in the MAC layer to improve the end-to-end performance. To reach this objective, we propose a simple greedy algorithm to schedule appropriately both the beacons and the active parts of the superframes.

We focus here on multihop wireless sensor networks where measures are collected by a sink, with a convergecast traffic pattern, and where nodes and the sink are static. However, topology changes may occur because of the radio link instability. In the same way some nodes may also be inserted in the network or may run out of energy. Consequently, we must construct a robust structure, i.e. the network shouldn't become disconnected because of one unique node failure.

The contributions of this paper are fourfold:

(1) we propose to maintain a new cluster-Directed Acyclic Graph structure at the MAC layer. Several parents are chosen to optimize both the robustness and the delay;
(2) we propose to increase the multihop network capacity by using the Beacon-Only Period jointly with a proper superframes scheduling while bandwidth waste is minimized;
(3) we present a simple and yet efficient localized scheduling scheme that assigns collision-free slots in a self-stabilizing manner;
(4) we evaluate the performance of RPL executed above this modified IEEE 802.15.4.

2 IEEE 802.15.4

IEEE 802.15.4 proposes a PHY and MAC layer in low-rate wireless personal area networks [1]. The devices can be classified according to their available resource and capabilities into Full-Function Devices (FFD) or Reduced-Function Devices (RFD). The FFD fully participate to the network, relay frames, and accept new nodes to associate with the network. On the contrary, RFD are

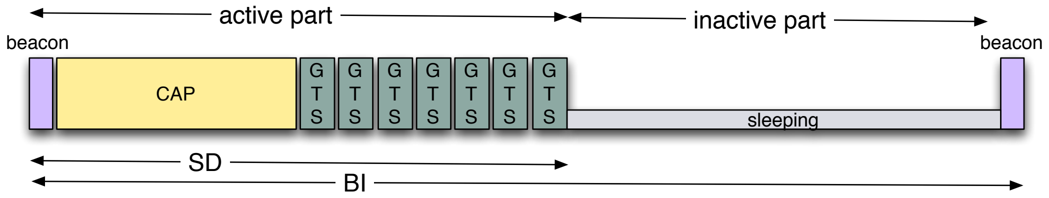


Fig. 1. Superframe structure in IEEE 802.15.4

very constrained nodes and constitute leaves: they sleep most of the time, and they turn-on their radio just to receive and transmit their own frames. IEEE 802.15.4 designates FFD as *coordinators*. Obviously, the direct transmission between two RFD is impossible: packets must be forwarded through a FFD.

2.1 Medium Access

IEEE 802.15.4 proposed two mutually exclusive modes: *non-beacon* and *beacon-enabled* modes. Without beacons, all the nodes use a CSMA-CA approach to transmit their frames. To enable very constrained nodes (RFD) to save energy, IEEE 802.15.4 implements an indirect transmission mode: a node cannot transmit its frames directly, it must wait a solicitation from the receiver. Consequently, a FFD must stay awake since it may receive a request from a neighbor at any time. In single hop topologies, all the nodes except the PAN coordinator may be RFD to save energy. However, in multihop topologies, the devices must forward the traffic. Consequently, they mud be FFD and cannot at all sleep in the non-beacon mode.

In *beacon-enabled* mode, IEEE 802.15.4 adopts the concept of superframes to implement a low duty-cycle mode (Figure 1). Each coordinator periodically sends a `beacon` including control information. During the active part of the superframe, any *child* may transmit a packet to the coordinator. In multihop, a node is both a child during the active part of its parent (uplink) and a coordinator for its own active part (downlink).

Just after having received a `beacon`, children may start a transmission during the Contention Access Period (CAP) using the slotted CSMA-CA. Children may reserve during the CAP a Guaranteed TimeSlot (GTS), dedicated for their periodic transmissions, at the end of the active part.

The active part of the superframe lasts for a `Superframe Duration` (SD), while beacons are transmitted every `Beacon Interval` (BI). When the superframe duration has finished, the coordinator may turn-off their radio to save energy: children cannot transmit frames anymore. The Superframe Duration (resp. Beacon Interval) are defined through the Beacon Order (resp. Superframe Order) values, according to the following relation:

$$SD = aBaseSuperFrameDuration * 2^{SO} \tag{1}$$
$$BI = aBaseSuperFrameDuration * 2^{BO} \tag{2}$$

By adjusting the BO and SO values, we can obtain a tradeoff between network capacity and energy savings. For instance, a duty cycle of 1% can be obtained if $SO = BO - 7$ ($2^{-7} < 1\%$). Dynamic solutions to adjust the duty-cycle ratio exist in the literature for IEEE 802.15.4 (e.g. [12])

The nodes turn-off their radio when they do not participate to any active part. In the same way, children may also turn their radio off during backoffs, waking up only for the CCA and for the transmission. A node is idle only when it waits for a `beacon`, an `acknowledgement` or a `data frame` in the indirect mode.

2.2 IEEE 802.15.4 Topology

An IEEE 802.15.4 network contains a PAN coordinator and a set of devices characterized by a limited transmission range and a limited quantity of energy. The PAN coordinator serves as a WSN gateway to the Internet and as well, as the primary controller of the network.

The IEEE 802.15.4 working group has proposed to support three different network topologies (Figure 2):

peer-to-peer: a node may communicate with any neighbor, the structure being decentralized. A routing protocol may enable multihop communication, using P2P transmissions at the MAC layer;

star: the PAN coordinator (a designated FFD) is in the radio range of all other nodes. A node forms a *branch of the star* and can communicate only with the PAN coordinator. Single hop transmissions are in this case sufficient for communication;

cluster-tree: presents a generalization of the star topology for multihop communication, enabled at the MAC layer. The coordinators (FFDs) of different clusters (stars) form a tree, rooted at the PAN coordinator. Traffic towards or from the PAN coordinator is forwarded by the coordinators.

Except for the peer-to-peer topology where a node can communicate with any neighbor, the star and cluster-tree topologies require from a node to be associated with one coordinator before transmitting packets. IEEE 802.15.4 implements a kind of topology control, where a node selects the neighbors with which it will communicate (its parent and children in the cluster-tree). A node needs to search for available coordinators by performing either active or passive discovery:

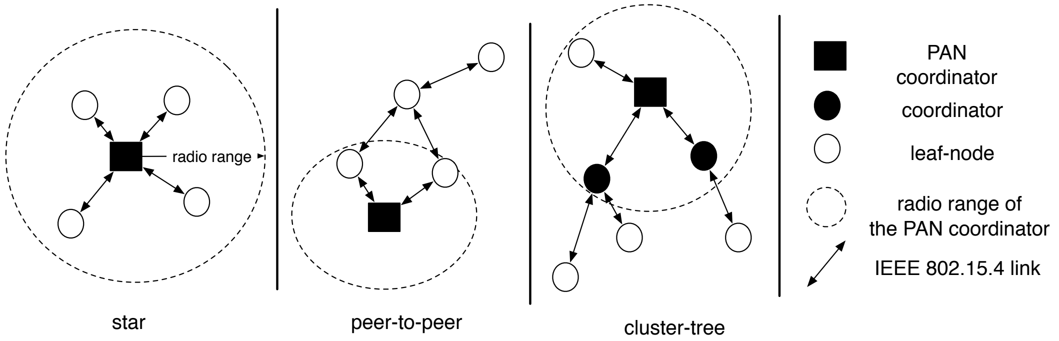


Fig. 2. Different topologies proposed in IEEE 802.15.4

- **Active scan:** a node is required to explicitly transmit `beacon-requests` to potential coordinators. The packets may collide with other data packets and with `beacons`. Besides, the emitter is never sure that the neighboring coordinators are awake, requiring to retransmit periodically the `beacon-requests`.
- **Passive scan:** a node scans through all available channels: it simply listens for incoming `beacons`. When a `beacon` is received, the source is inserted in the list of possible parents.

After having completed a discovery procedure, a node has a list of all available coordinators, and thus, can initiate an association procedure to the selected coordinator.

The association procedure requires a 6-way handshake: first, a node sends an `association-request` during the CAP, acknowledged by the coordinator. Afterwards, a node waits `macResponseWaitTime` before transmitting a `data-request`. Finally, the coordinator replies with an `ack` followed by an `association-reply` attributing a short address (16 bits) to the node. A node completes the association by acknowledging the `association-reply`. The association may use several superframes if $SD <$ `macResponseWaitTime`.

Meng et al. [22] propose an optimized association scheme. A scan is stopped as soon as one of the discovered PAN coordinators is estimated worthy to associate with. The association scheme itself excludes the *data-request* primitive and *macResponseWaitTime* to finally result in accelerated convergence time by 90%. Karowski et al. [17] have proposed an interleaving heuristic to discover on average more quickly new coordinators by using small BO values. Zigbee [31] proposes to couple the association procedure with an address assignment scheme: addresses being hierarchical, routing is simplified.

A node becomes an orphan when it looses synchronization with its associated parent i.e. when it misses 4 consecutive beacon frames. An orphan either reassociates with the previous parent or tries to find a new one. It may stay disconnected for a long time, especially when a node is running on a low duty-cycle. The cluster-tree is not robust since a node has a single parent. A node must select a stable parent with a good link quality since it greatly impacts performance.

Moreover, the standard does not specify what coordinator a node should choose to associate with, just the way information is exchanged during the association. The properties of a cluster-tree when a node associates with the first available [9] and a random parent among those satisfying the link quality[10] have been studied recently. Nevertheless, only few attention has been given to determining the required characteristics of the cluster-tree and what algorithms could obtain them. We will address this problem in the rest of this paper.

Choosing the right neighbor is related to the topology control problem. A node may select its neighbor to reduce its transmission power for broadcast [7] or to deal with time varying links [21]. Our solution described below may integrate any sophisticated depth metric to achieve such objectives.

2.3 Extension of IEEE *802.15.4 for multihop WSN*

For a node, the active part of its parent is designated as *incoming* and the active part maintained by the node itself as *outgoing*. The standard suggests that the active parts of one child and its parent are interspaced by *StartTime*. If the *StartTime* value is constant, coordinators with the same depth start synchronously their active part. Practically, `beacons` collide, making the protocol inefficient.

There are two main approaches in the literature to reduce the number of collisions. In the *Beacon Only Period* (BOP) solution, nodes implement a TDMA approach to send their `beacons`: at the beginning of each active part a few slots are dedicated to `beacons` [19]. While collisions are avoided during the BOP, data frames may still collide in the second part since the data part may overlap. Simulations showed that performance quickly degrades if hidden terminals are frequent [4].

A second solution consists of using variable *StartTime*: two nodes that have the same parent should for instance not use the same *StartTime* so that their active parts do not overlap [2]. If we assume that all the nodes use the same BO and SO values, finding the adequate *StartTime* for all of them is equivalent to scheduling the active parts with a TDMA approach. Villaverde *et al.* [28] have experimentally proved that this approach leads to the best performance.

Koubaa *et al.* [19] proposed a centralized algorithm to schedule the active parts with a variable superframe duration (which corresponds to a classical knapsack problem). Muthukumaran *et al.* proposed a greedy distributed algorithm to

algo	beacons	active periods	algorithm	Remark
IEEE **802.15.4**	none	parent and child inter-spaced by *StartTime*	none	both data and beacon collisions frequent
BOP	TDMA of beacon slots	shared network wide	distributed (first free)	data collisions remain
variable *StartTime*	none	TDMA of active periods	centralized, distributed (first free)	efficient distributed algorithm to be proposed

Table 1
Comparison of techniques to reduce the number of collisions in IEEE 802.15.4-2006

pick the first free slot [23]. This algorithm creates a burst of collisions when the children of the same parent choose to start simultaneously their active parts. Rhee *et al* [25] presented a distributed slot assignment for a TDMA MAC in wireless sensor networks. The authors use a localized greedy algorithm to pick a free slot for the transmission. We were inspired by this approach in assigning slots in a distributed way.

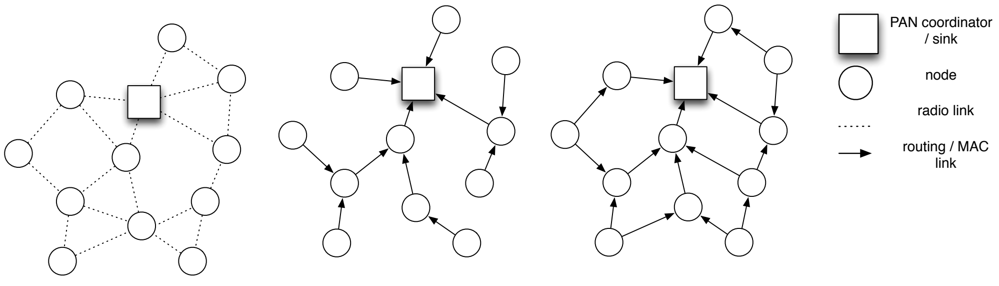
Table 1 references the behavior of the different approaches to limit the number of collisions with IEEE 802.15.4-2006.

2.4 Routing on top of IEEE *802.15.4*

RPL [30] is the emerging standard for routing in Wireless Sensor Networks. It creates a Destination Oriented-Directed Acyclic Graph (DODAG) rooted at the sink (Fig. 3c). A Directed Acyclic Graph (DAG) [27] is an oriented graph with no cycle. If we place the PAN coordinator as the root of this DAG, we maintain several paths toward the PAN coordinator while forbidding any routing loop.

Each node broadcasts a DIO (DAG Information Object) including its distance to the root of the DAG according to a given metric (e.g hop count, link quality,delay, or jitter). Then, each node executes a distance vector algorithm to find a set of neighbors closer to the root than itself: they become its parents. RPL also proposes a mechanism for fast route repair when a transient loop is detected.

While RPL has been implemented both in TinyOS and Contiki and thoroughly evaluated [18], it is rarely executed over a low-duty cycle MAC. To the best of our knowledge, RPL was not even evaluated when functioning with the



(a) radio topology (b) cluster-tree (IEEE 802.15.4) (c) DODAG (RPL over radio topology)

Fig. 3. Topology control in a IEEE 802.15.4/ RPL network

	structure type	nb. of parents	metric type	forwarding choice
IEEE **802.15.4**	cluster-tree	1	none	single associated parent (static)
IETF RPL	DODAG	up to 3	hop, ETX, delay, jitter	single preferred parent (dynamic)

Table 2
Comparison of IEEE 802.15.4 and IETF RPL vis-a-vis topology control and forwarding choice

beacon-enabled mode of IEEE 802.15.4.

Recently, several routing approaches advocate the usage of multiple paths in Wireless Sensor Networks. Villaverde et al. [29] proposed to select the best route among a set of available paths to meet some QoS requirements for industrial applications.

3 Problem Statement

As introduced previously, IEEE 802.15.4 proposes to construct a cluster-tree on top of a radio topology (Fig. 3b). If we execute RPL over the beacon-enabled mode of IEEE 802.15.4, it must use the cluster-tree structure. Thus, RPL cannot fully exploit the mesh radio topology: one single path exists toward the PAN coordinator.

We rather propose to modify IEEE 802.15.4 so that RPL can exploit a meshed redundant topology, creating a proper DAG (Fig. 3c). We aim at modifying the MAC layer so that each node maintains several parents.

This multipath structure is vital to implement QoS routing: different routes may present different routing metrics and would avoid maintaining several separated DAG, thus, decreasing control traffic. Besides, the DAG structure also improves robustness: the network should keep on functioning even if a few radio links are broken. The original cluster-tree structure of IEEE 802.15.4 does not guarantee such feature.

We propose to build and maintain a new cluster-Directed Acyclic Graph (cluster-DAG) structure at the MAC layer. A node establishes a bidirectional link by associating explicitly with several parents.

Since each node has to maintain its own superframe, we must also avoid beacon collisions among coordinators and inside the data part of each superframe. Thus, we propose to further improve the IEEE 802.15.4 multihop operation. We implement both a Beacon-Only Period and an efficient superframe scheduling algorithm. By superframe scheduling, we mean scheduling the active parts of the different nodes. We aim at avoiding beacon collisions while reducing the bandwidth waste due to unused superframes. Extensive simulations confirm the efficiency of this superframe organization.

4 New topological structure: cluster-DAG

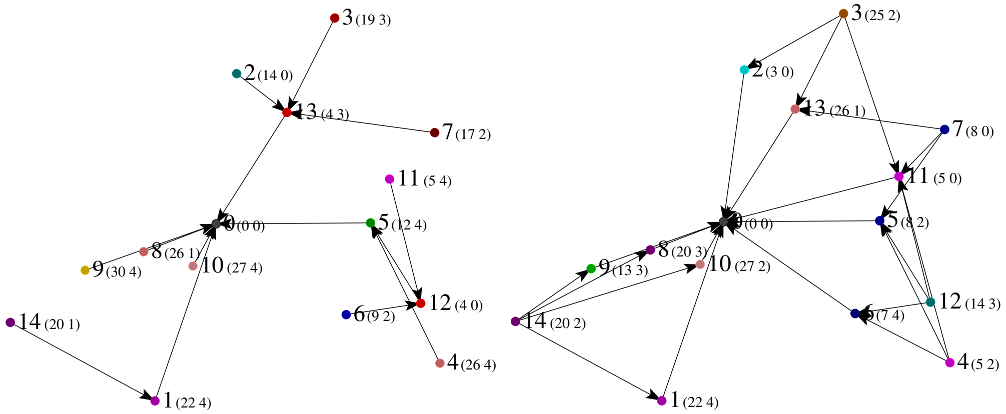
We aim at running RPL on top of the IEEE 802.15.4. The cluster-tree structure imposes a single available route on RPL. We rather propose to construct a cluster-Directed Acyclic Graph in the IEEE 802.15.4 layer.

Let us consider Figure 4. Each vertex is labeled $X(Y, Z)$, where X denotes the node id. We can remark that in the DAG version, a node has several parents, thus several routes toward the PAN coordinator (0) exist. We can remark that the cluster-DAG structure permits to introduce more redundancy, even in a such simple topology with a low node degree.

4.1 Multiple parent association

We chose to re-use the passive coordinator discovery procedure of IEEE 802.15.4. On the other side, the IEEE 802.15.4 standard does not actually define how to choose a parent.

We decide to use the beacon frames to piggyback the depth of the coordinator in the sense of some metric. A depth may denote the hop distance from the PAN coordinator. However, minimizing the hop distance leads to selecting of-



(a) Cluster-tree (b) Cluster-DAG

Fig. 4. Illustration of the topology constructed by the IEEE 802.15.4– original cluster-tree and cluster-DAG version

Variable	Definition
$\mathcal{P}_{assoc}$	set of already associated parents
$\mathcal{P}_{on-going}$	set of already and on-going associated parents
$depth(P)$	depth of parent P in sense of some metric toward the PAN coordinator)
$metric(N, P)$	metric associated to the link (N,P) (e.g. 1 if we consider number of hops)
$depth(N, P)$	depth of node N when using the parent P NB: $depth(N, P) = depth(P) + metric(N, P)$
Δ	constant limiting the sub-optimality deviation

Table 3
Notation used for parent selection algorithm

ten bad radio links [11]. Thus, we chose rather to use the cumulative Expected Transmission Count (ETX) to denote the depth.

To estimate the ETX, a Node N monitors each possible parent P: it counts the number of `beacons` it receives from this parent, and from the periodicity, interpolates the packet delivery ratio (PDR). We assume here the radio link is symmetrical, and we consider ETX is equal to the inverse of the packet delivery ratio from the parent to the node. Any other more sophisticated metric may be used here to select the best parent.

Finally, a node engages the first association procedure with the parent offering the smallest depth. Instead of stopping the association, a node keeps on searching and associating with new parents. Nodes stay awake and keep on listening to incoming `beacons` to find alternative candidates.

Let us consider node N receiving a `beacon` from node P. We will use the notation described in Table 3. N applies the following rules when associating with more parents:

selection based on depth: N is not yet associated with P, neither P is a child of N. The depth of P is strictly inferior to the depth of all already associated parents of N plus depth threshold Δ:

$$depth(N,P) < min_{p \in \mathcal{P}_{on-going}} \left(depth(N,p)\right) + \Delta \tag{3}$$

N engages the association procedure and flags the parent P as `on-going`. We compare the depth for both on-going and already terminated associations. In this way, we avoid associating with a new parent that will soon become suboptimal when some other on-going association terminates.

disassociation: a node N is associated with parent P, but the depth of P is superior to the depth of at least one associated parent plus Δ threshold:

$$depth(N,P) \geq min_{p \in \mathcal{P}_{assoc}} \left(depth(N,p)\right) + \Delta \tag{4}$$

We consider only already associated parents to judge if an old parent became suboptimal. Node N engages a dissociation with P.

Finally, we adopt an optimistic cumulative metric. A node announces a route with the minimal cost to reach the sink. Node N piggybacks in its `beacons` the minimal depth among already associated parents:

$$depth(N) = min_{p \in \mathcal{P}_{assoc}} \left(depth(N,p)\right) \tag{5}$$

To avoid the creation of loops, we must carefully choose Δ. To create a DAG, Δ must be inferior to the minimal metric for a radio link. If we use either ETX or the hop distance to measure the depth, the minimal metric for a link is 1. Therefore, we fixed $\Delta = 1$ in the simulations.

4.2 Energy overhead due to multiple parents

In IEEE 802.15.4, a node consumes most of its energy as coordinator: it has to stay awake during the whole active part (e.g. 61.44 ms for $SO = 2$).

A node following a superframe just requires to wake-up and to receive the `beacon`. To deal with clock-drifts, a follower must reserve a guard time before the expected `beacon` transmission. A typical clock drift is equal to 10 μs per second [24] (10 ppm), and the `beacon` transmission lasts about 100 μs. Finally, following a new parent costs a node 0.18% more energy. We consider that this energy overhead could be neglected.

4.3 The complexity of parent selection

Maintaining several parents increases the memory requirements only linearly with the number of parents. Since a node saves a very limited number of information about each parent [2], we consider this memory constraint acceptable.

The parent selection rule is executed after a node receives a `beacon`. The conditions are the same as the single parent case: we just replaced the strict inequality by a non strict inequality, and insert the Δ constant (eq. 3 & 4). Thus the computational complexity remains unchanged.

5 Beacon / Superframe Collision Avoidance

Within the cluster-DAG structure, a node is able to associate with several parents. However, we must carefully schedule the beacons and superframes to avoid collisions. Else, we would increase the number of disassociations, leading possibly to a network partition. If all nodes are synchronized, their superframes may be scheduled without overlap. For the sake of simplicity, we may interchangeably use the *active part of the superframe* and the *superframe*.

We adopt the following organization combining both the superframe scheduling and Beacon-Only Periods (BOP) (cf. Fig. 5):

- a Beacon-Only-Period is reserved at the beginning of each superframe with $n_{bop-slot}$ slots. When several coordinators interfere, they share the same active part. However, at most one of them should have traffic to receive, that is to say, an associated child.
- we implement a TDMA approach to organize the superframes. As mentioned in the standard, the BO value must be uniform in the network. The number of superframe slots (denoted $n_{sf-slot}$) is equal to 2^{BO-SO}. Each slot contains the active part of a superframe.

Such a combined approach offers the following features:

(1) two beacons should not collide;
(2) a coordinator without children should not reserve much bandwidth for its own use: no `data` packets are transmitted during its active part.

---

[2] typically, a node must save for each parent its IEEE 802.15.4 short address (2 bytes), depth (1 byte), and 2 flags (*it_has_children*, *I_am_associated*)

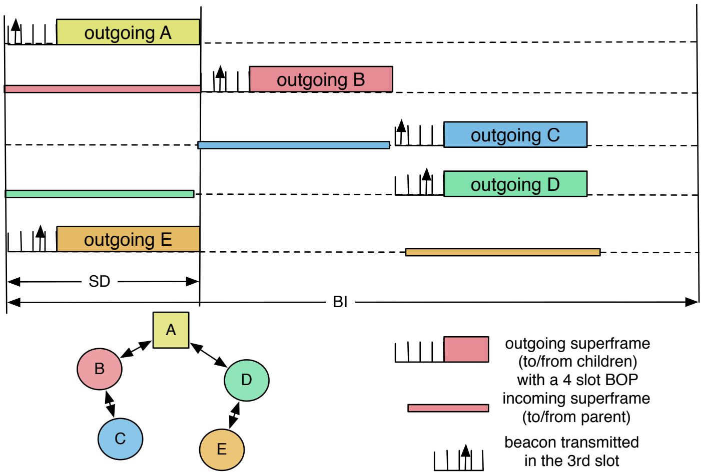


Fig. 5. Proposed superframe collision free organization: we combine Beacon-Only-Period with superframe scheduling

5.1 *Neighborhood discovery and maintenance*

To avoid collisions, a coordinator must collect information about its neighborhood. In particular, each node must maintain a list of interfering coordinators with their chosen active and BOP slot. Therefore, a node creates `hello` packets containing the list of its (interfering) neighbors at most k hops away with the following information:

- superframe slot (1 byte);
- BOP slot (4 bits);
- does this node has a child? (1 bit);
- depth (1 byte).

When a `hello` is received, the receiver updates its neighborhood table accordingly. When a node detects a change in the neighborhood, a `hello sequence number` is automatically incremented. In conclusion, each node maintains the list of k-neighbors and the BOP / superframe slots they use.

Furthermore, each coordinator transmits in its `beacons` its current `hello sequence number` (1 byte) so that each neighbor is able to detect a change occurred.

Finally, an `hello` contains the list of its (k-1)-neighbors with, for each neighbor:

- short address (2 bytes);
- depth in the cluster-DAG (1 byte);
- superframe slot (1 byte) and BOP slot (4 bits);

On this basis, we implement two methods for the k-neighborhood discovery:

active discovery: after having received the `beacon`, a node may detect that the `hello sequence number` of the source has changed since the last time. A node will trigger a `data-request` during the Contention Access Period to retrieve the new `hello`. The active method is recommended when a node detects a new neighbor and needs immediately information about it.

passive discovery: whenever the `hello sequence number` changes, the coordinator broadcasts the new `hello` after its next `beacon`. Thus, a node has just to wait for this `hello` packet during the coordinator's next superframe. The passive method is efficient when a node already knows the previous neighborhood table of a transmitter and has just to know when it changes.

Separating `beacons` and `hellos` solve several problems. First, we limit the overhead when the neighborhood is stable: no additional information is piggybacked in the `beacons`. Second, the `beacon` must entirely fit the BOP slot. Dealing with large densities would consequently require reserving more bandwidth for `beacons` and reducing the time dedicated to data exchanges. Third, using the `beacon` avoids to adapt the `hello` period with solutions like the trickle timer [20] or TAP [16].

5.2 Superframe slot assignment

A node must schedule its active part while avoiding collisions among both `beacons` and `data` packets. We propose two distributed strategies.

5.2.1 Random assignment

As random strategies often perform efficiently in wireless multihop networks, we propose a very naive and simple approach. A coordinator randomly selects one superframe slot not used by any parent.

Let n_{coord} be the number of mutually interfering coordinators and $n_{sf-slot}$ be the number of superframe slots. The probability that at least one collision occurs is (the birthday problem):

$$1 - \prod_{i \in [1..n_{coord}-1]} \left(1 - \frac{i}{n_{sf-slot}}\right) \tag{6}$$

Since the number of interfering coordinators increases with the density, collisions may quickly arise if (BO-SO) is too small (Fig. 6).

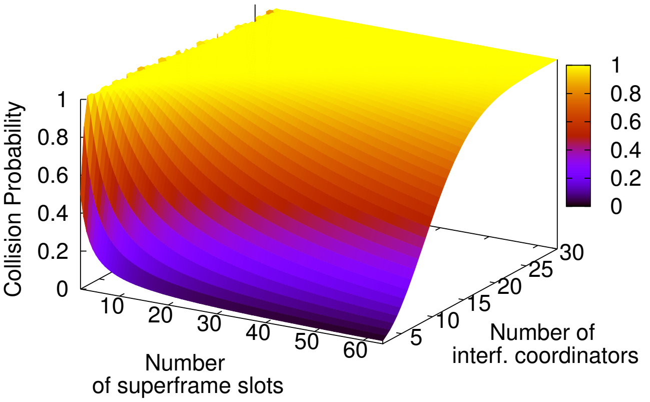


Fig. 6. Impact of the number of slots and interfering coordinators on the superframe collision probability

5.2.2 Greedy assignment

We also propose a greedy solution, assigning one superframe slot according to the load extracted from the neighborhood table. A node N extracts all the slots used by k-neighbors (interfering nodes) and applies the following rules:

(1) if several slots with no interfering coordinator exist, N randomly chooses one of them;
(2) else, if all slots are occupied, a node N tries to avoid a collision with a coordinator with children;
  (a) if N has at least one child, it chooses a slot that does not belong to a smaller ID coordinator with children. N also applies a conservative strategy: if the previously selected slot is free, it will maintain it;
  (b) Else, if N does not have children, N considers all the coordinators with children and all smaller ID coordinators without children:
    (i) N blacklists the superframe slots with more than $n_{bop-slot}$ participants from the list [3]
    (ii) N sorts the remaining slots according to the number of coordinators using it. The *best-slots* contain the lowest number of coordinators.
    (iii) N randomly chooses one of the *best-slots*. N should not apply a conservative strategy: since it does not have children, it can safely change its slot. Practically, randomization helps to accelerate the convergence. That is to say, other nodes are able to test the same slot for their own usage, without creating collisions.

---

[3] If a node has more than $n_{bop-slot} * n_{sf-slot}$ interfering coordinators, no superframe with less than $n_{bop-slot}$ coordinators may exist. Such a scenario is unrealistic: a scheduling solution without collision does not exist.

In conclusion, a node chooses to limit the collisions with interfering coordinators with children: they will surely have data packets to receive/transmit during the CAP. Collisions with other coordinators are solved by using different BOP slots. Besides, a node only considers interfering nodes with a smaller ID to avoid deadlocks. Even if a node N initially chooses the same slot as an interfering smaller ID coordinator, situation is perceived and corrected by N with the reception of new `hello`.

A coordinator continuously executes previously stated rules at the beginning of its superframe. If another slot should be used, a node engages the *handoff* procedure as follows:

- the coordinator transmits a `beacon` as usually, but piggybacks its new superframe slot;
- all children receive the `beacon`, update the superframe slot of their parent, and turn-off their radio for the rest of the slot. They will wake-up only for the next `beacon` of this coordinator.
- a node cannot associate with this coordinator (the current slot is not the one announced in the `beacon`). A node just updates its neighborhood table.

We can note that this procedure is only important for the coordinators with children. Others may change several times their slot without impact on the cluster-DAG.

5.3 BOP slot assignment

After having selected the same superframe slot, several coordinators may still interfere with each other. The Beacon-Only Period permits to solve this problem: the coordinators would only contend during the BOP and not during the CAP. More precisely, when at most one of them has children, our combined approach reduces the number of collisions, particularly frequent in presence of hidden terminals.

Each coordinator constructs the list of BOP slots and the number of interfering coordinators in each of them. It finally randomly chooses a free slot.

When a new BOP slot is selected, a coordinator must trigger a CCA before transmitting its own `beacon`. In this way, we avoid collisions among two interfering coordinators choosing the same BOP slot in two consecutive superframes.

A coordinator without children might change its BOP slot for different superframes. Nevertheless, we forbid selecting a BOP slot chosen by a coordinator with children. In this way, we can detect collisions between coordinators without children. Practically, the number of coordinators in the same superframe slot is low since it tries to minimize the number of coordinators per slot.

5.4 Self-Stabilization

An algorithm is self-stabilizing if it converges to a legal state in a finite number of steps regardless of the initial state [13]. We denote by the legal state a cluster-DAG in which no pair of interfering coordinators with children has the same superframe slot and in which no pair of interfering coordinators selects the same BOP and the superframe slot. In this case, we do not have any `beacon` or data collision between different coordinators, but rather between children participating to the same superframe of the same coordinator.

Clearly, the random assignment is not self-stabilizing: a pair of interfering coordinators may select the same slot. The greedy approach is self-stabilizing: when a conflict is detected, the coordinator with a larger ID will change its decision.

More formally, we assume that a pair of interfering coordinator will detect a collision. If the interfering range is large, we just have to increase the k-neighborhood discovery: a coordinator considers that it interfers with any k-neighbor. We would over-estimate interference, but a collision will be detected. In our simulations, we have chosen $k = 2$.

If the collision occurs within a k-neighbor ($k > 1$), the intermediary nodes would eventually exchange their neighborhood table. The pair of colliding coordinators becomes immediately aware of this collision. Subsequently, the coordinator without children or with a larger ID will change its slot choice.

Otherwise, the following illegal cases can occur between two 1-neighbors:

(1) a pair of coordinators with children has the same superframe slot and a different BOP slot: each of them receives the `beacon` and a `hello` packet from the other one. The coordinator with the largest ID will consequently apply the algorithm (choosing another slot);
(2) a pair of coordinators shares the same superframe and BOP slots: `beacons` and `hellos` collide, and the coordinators may be unaware of each other since they do not have a common neighbor. In this case, neighbors cannot associate with the coordinators due to the `beacon` collisions. Thus, the coordinator that last selected this BOP slot will not have any children. The coordinator without children will randomly select another BOP slot removing the collision in the next superframe;
(3) more than $n_{bop-slot}$ coordinators are present in a superframe. At least one BOP slot contains more than two coordinators creating `beacons`

> collisions. `hellos` are also lost similarly making the pair of coordinators unaware of each other. However, coordinators without children randomly change their BOP slot: they will receive the `hellos` during the next superframes. Consequently, they will detect that the superframe slot has too many coordinators and at least one of them will select another one. Since we assume that we have a sufficient number of superframes and BOP slots to avoid collisions between `beacons` (otherwise, the network is not correctly configured), the problem will be solved.

In summary, at least one of the coordinators detects the illegal state and changes its decision choosing a legal BOP/superframe slot. Finally, the system converges to a legal state.

5.5 Complexity of BOP/superframe slot selection algorithm

The algorithm to select the superframe and BOP slots has to be executed once every Superframe Duration (SD), at the beginning of the superframe slot.

The computational complexity of the superframe slot selection is linear with the number of k-neighbors. More precisely, the algorithm consists in counting the number of k-neighbors for each superframe slot. Let N_{slots} be the number of slots and Δ_k be the number of k-neighbors. The first step of the algorithm tries to find an empty slot ($O(N_{slots})$). If no empty slot is found, the algorithm selects the slot with the smallest number of nodes. Since we have at most Δ_k non empty slots, the overall complexity is finally in $O(\Delta_k)$.

The BOP slot is selected by looking for an empty BOP slot in the k-neighborhood table. Thus, its complexity is in $O(\Delta_k)$.

Finally, the memory requirements also increase for our algorithm, since we must save a k-neighborhood table. For each neighbor, a node must save its superframe (1 byte) and BOP slots (4 bits), its short address (2 bytes), its distance in hops (4bits), and if it has children (1 bit). We consider this memory size remains acceptable.

6 Exploiting the cluster-DAG structure

The original IEEE 802.15.4 forwarding scheme exploits a single available parent. It can lead to a single point of failure and potential packet drops due to a disconnection. Also, forwarding the integrality of traffic to a single parent can easily lead to congestion and performance degradation.

On the other hand, the RPL forwarding scheme operates on a larger parent set (up to 3 parents). Traffic is always forwarded to a preferred parent offering the best performance regarding a chosen metric. Whereas, the alternative parents serve to replace the preferred parent in the case of unavailability or performance degradation. The alternative parents render the DODAG structure more robust. Nevertheless, when the preferred parent is stable (the path quality does not change), it is always used for forwarding.

As an alternative, we propose to further exploit the advantages of the cluster-DAG. A node deploys an opportunistic anycast strategy: the node picks a packet from its buffer regardless of the current parent. Such a straightforward approach manages to distribute the traffic among more parents. Finally, it may lead to performance improvements—increased PDR and decreased delay.

7 Performance evaluation

Radio range	30 m	SO	2
Interference range	60 m	BO	7
avg. nb. of neighbors	8	path loss	1.97
nb. nodes	60	standard deviation	2.0
Inter packet time	100s	Pr(2m)	-61.4dBm
BOP slots	4	depth metric	ETX

Table 4
Default values used in the simulations

We have used WSNet, an event-driven simulator for large scale wireless sensor networks ([14]) and the beacon-enabled mode of IEEE 802.15.4 [3]. The simulator has been already thoroughly evaluated [6].

We focus on static topologies of nodes with one sink, and with a convergecast traffic pattern. To model radio links with variable radio link quality, we used the path-loss shadowing model calibrated with the scenario FB6 (indoor real deployment) [8].

The default simulation parameters are represented in Table 4. We have simulated a duty-cycle between 1% (2^{2-9}) and 25% (2^{2-4}). We have plotted 95% confidence intervals.

We have implemented 3 solutions for comparison:

(1) 802.15.4: the slots of one coordinator and its parent are consecutive;
(2) random strategy (Section 5.2.1);

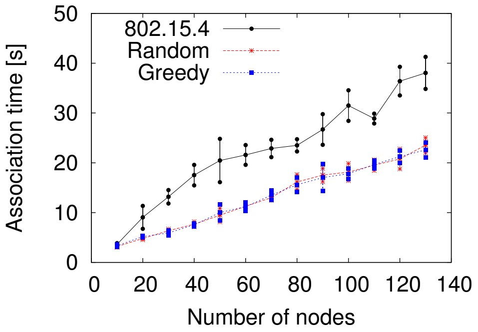


Fig. 7. Time until the last node associates with the cluster-DAG

(3) greedy scheduling (Section 5.2.2).

We simulate periodic convergecast traffic: a node generates one packet for the PAN coordinator every $T_{interpk}$. Each node maintains a FIFO (First In First Out) buffer. A packet is pulled from the buffer when the node is in the idle state during the CAP of the superframe of its parent. A node periodically removes packets that exceeded their timeout (`macTransactionPersistenceTime` as defined in the IEEE 802.15.4 standard).

We have implemented two different routing strategies:

- unicast: a node forwards all its traffic to a single parent. If a node has several parents, it selects always as next hop its *preferred parent* (parent with the best *depth* metric);
- anycast: a node by default applies an opportunistic anycast forwarding (it sends the packets to the first available parent).

We have measured the following metrics to evaluate the performance of the network:

- packet delivery ratio: the ratio between the number of transmitted packets and the number of received packets
- end-to-end delay: the time between the packet generation instant and its reception by the PAN coordinator (i.e. the sink);
- BOP/superframe collision ratio: the ratio of coordinators that have an interfering coordinator sending a beacon at the same instant;
- number of parents: the average number of parents for each node;
- association time: the time until the last node associates with the cluster-DAG.

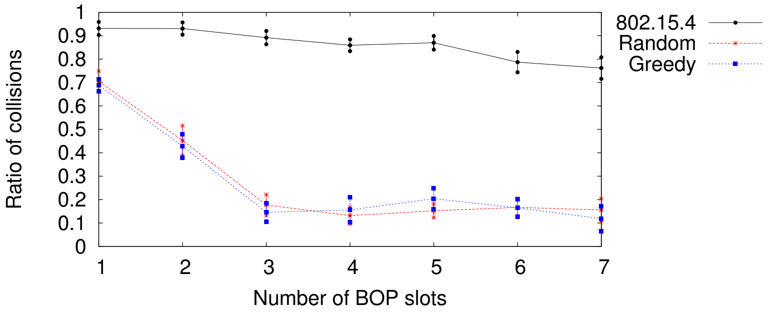
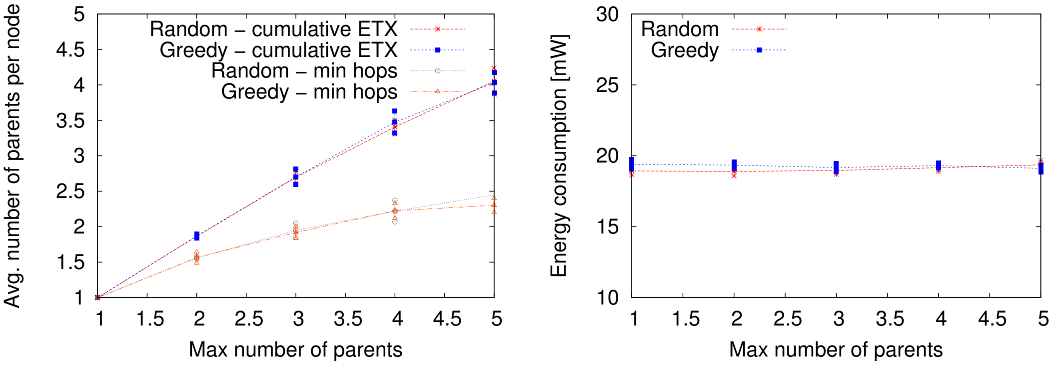


Fig. 8. Impact of the number of BOP slots on performance



(a) Average number of parents per node

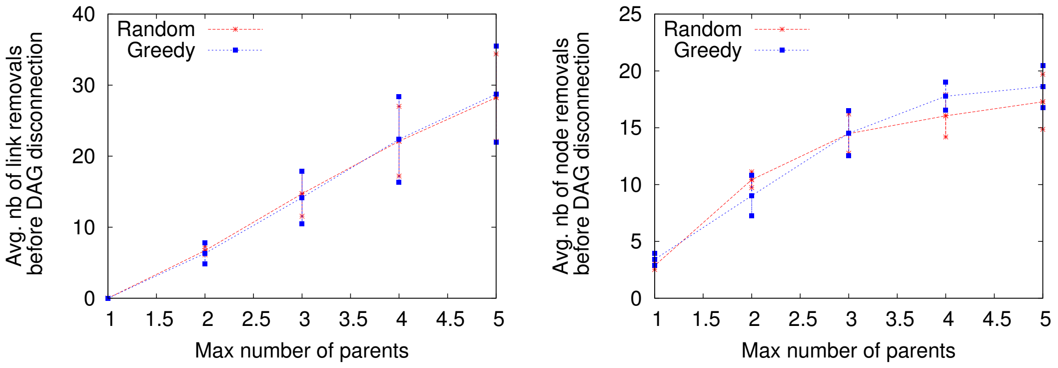
(b) Energy Consumption

Fig. 9. Impact of the number of parents on performance

7.1 Cluster-DAG properties

Then, we have measured the time until the last node becomes associated, i.e. it has a valid parent and it gets a short 16 bits address (Fig. 7). The number of nodes increases while maintaining the density constant. The association time increases slightly for the random and greedy strategies: the larger network diameter means that the last node will wait longer to have an associated neighbor. However, the association time for the IEEE 802.15.4 strategy increases: collisions quickly impact the convergence time.

Finally, we have measured the impact of the number of BOP slots on performance (Fig. 8). We can remark that increasing the number of BOP slots reduces the number of collisions (i.e. coordinators that use the same superframe and BOP slots). Thus, we use 4 BOP slots in the rest of the simulations.



(a) Link Connectivity

(b) Node Connectivity

Fig. 10. Connectivity of the Cluster-DAG

7.2 Impact of multiple parents

We have first evaluated the structural properties of the cluster-DAG. We have measured the real number of parents according to the number of maximum parents a node is allowed to select (Fig. 9a). As expected, the redundancy increases when the maximum number of parents is larger.

We have also measured the energy consumption (Fig. 9b). We can verify that both random and greedy strategies consume almost the same amount of energy. Besides, having more parents does not really impact the energy consumption. It actually seems that having more parents with the greedy strategy reduces the energy consumption: balancing the load among efficient parents may reduce the number of collisions.

Finally, we have measured the node (respectively, link) connectivity (Fig. 10). We randomly remove one node (respectively, link) and verify if the cluster DAG is still connected: a path exists to the PAN coordinator. We express the connectivity as the number of nodes (respectively, links) that can be removed before the disconnection. The value is averaged over 50 random sets of removed nodes (respectively links). This connectivity metrics is related to the ability of the network to cope with e.g. a node running out of energy or a variable radio environment. Without a surprise, increasing the number of parents really helps to improve connectivity. For example, with at most 3 parents, 15 links or 14 nodes have to be removed to create a partition in the network. We may also verify that the cluster-tree (number of parents = 1) is particularly weak: because of the tree structure, any link is vital. Besides, only the leaves of the cluster-tree can be removed without creating a partition.

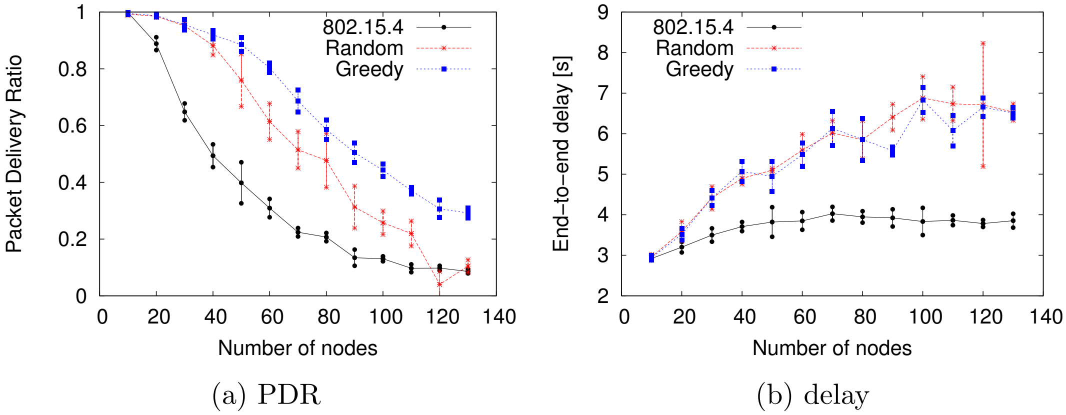


Fig. 11. Scalability of slot attribution algorithms

7.3 Scalability

We have also evaluated the scalability of the proposed scheme. We have observed the impact of the increased number of nodes on performance.

We have first measured the packet delivery ratio – PDR (Fig. 11a). The PDR decreases when more nodes are present: the global traffic also increases creating more collisions. Besides, the average route length also increases, decreasing the probability that the packet reaches the destination. However, the greedy strategy outperforms the others. While IEEE 802.15.4 delivers only 30% of the packets with 60 nodes, the random strategy delivers 55% of the packets and the greedy strategy 85%.

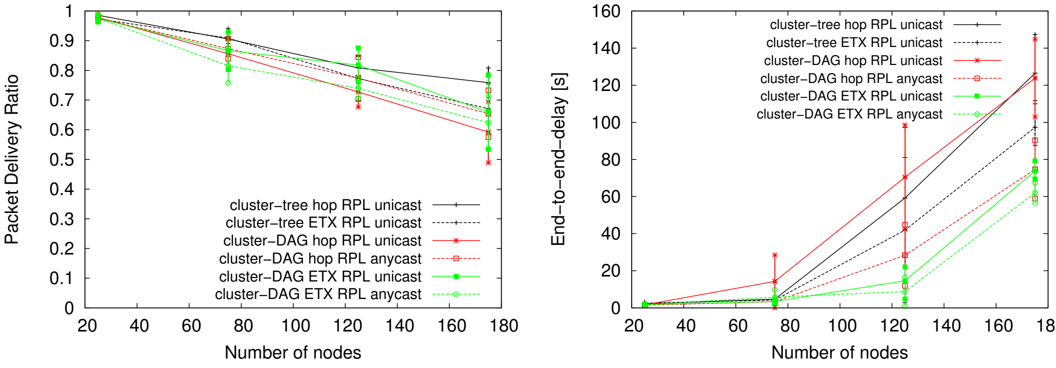
We have also measured the delay (Fig. 11b). The delay for the original strategy is lower, but mainly because most packets are dropped. For the other strategies, the delay increases almost linearly with the number of nodes, which is a quite good property.

7.4 Routing with RPL over a cluster-DAG

Finally, we investigated the impact of the forwarding scheme and underlying topological structure on the routing performance. To isolate the impact of the topology structure and the routing protocol, we verified that the scheduling leads to a collision-free schedule.

We compare our 2 different routing scenarios (unicast and anycast) with the following *depth* metrics used for topology construction:

min-hops : the depth uses the hop count to the PAN coordinator;



(a) Packet Delivery Ratio

(b) End-to-end delay

Fig. 12. Scalability of different routing strategies

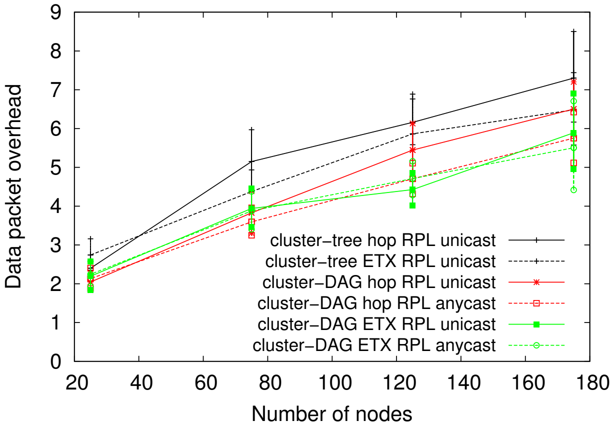


Fig. 13. Routing energy (data packet) overhead

ETX : the depth uses the cumulative ETX, estimated through the beacon packet delivery ratio, as described in section 4.1.

We first measured the Packet Delivery Ratio and the end-to-end delay (Fig. 12). Surprisingly, the differences between the different schemes are limited. Unburden with buffer size limitation, routing over the cluster-tree optimized for hop count achieves the highest PDR. RPL anycast routing over a cluster-DAG experiences a slightly lower PDR performance. It is due to the simultaneous use of alternative parents of lower quality.

Nevertheless, the RPL anycast forwarding over a cluster-DAG shows its advantages in terms of delay. Indeed, anycast distributes the load among different parents: we reduce the probability that a single path forwards most of the traffic. Thus, reducing the queue sizes often impacts positively the end-to-end delay. In other words, we achieve the same reliability with a slightly lower number of packets.

We also measured the routing energy (data packet) overhead. We express it in the average number of necessary packet retransmission to successfully deliver a data packet to a sink (Fig. 13). Results are averaged over all successfully delivered data packets.

We can observe in Fig. 13 that cluster-DAG structure outperforms a simple cluster-tree in terms of energy overhead. A positive effect of maintaining up to 3 parents in total can be seen in overall lower energy overhead. We avoid congestion by distributing the data traffic among all available parents. However, we can notice that overhead increases slightly when RPL unicast is used instead of RPL anycast over the cluster-DAG based on the hop metric. In conclusion, a multi-path RPL anycast routing protocol incurs in total less packet transmissions when used with cluster-DAG, leading to a lower energy consumption at the end. With 160 nodes, a cluster-tree (min hops/unicast) requires 20% more packets than a cluster-dag (min hops or ETX/anycast).

8 Conclusions and Perspectives

We have proposed to modify the topology of IEEE 802.15.4 by adopting a Directed Acyclic Graph structure. This cluster-DAG structure permits to create alternative path to the root. In particular, a routing protocol like RPL is able to exploit a redundant topology. We also provided simple greedy scheduling algorithms to schedule the active parts of the superframes and the beacons, adapted to this DAG structure. This solution avoids the collisions of both beacons and data frames while limiting bandwidth waste.

Simulation results demonstrate the interest of this DAG structure to make the network more robust: the suppression of a radio link and/or a node has a smaller impact on the performance. Besides, we also proved that RPL is more efficient on this cluster-DAG, to reduce the delay. The number of transmitted packets is also lower, decreasing consequently the energy consumption since a lower duty-cycle ratio may be used to forward the same amount of traffic.

In the future, we plan to evaluate experimentally our solution and measure its scalability. We conjecture that our algorithm that reactively detects collisions will be robust under complex interference patterns. We also plan to investigate a closer integration of RPL and IEEE 802.15.4. In particular, the metric used to construct both the MAC and routing DODAG should be the same, else we would obtain a suboptimal routing structure. Moreover, guaranteeing a certain Quality of Service would require a close interaction between both layers, for the queue management, the residual bandwidth evaluation, the load balancing.

References

[1] IEEE 802.15.4-2006 standard, http://www.ieee802.org/15/pub/04.

[2] IEEE 802.15 WPAN task group 4b (tg4b), http://www.ieee802.org/15/pub/TG4b.html (September 2004).

[3] N. Abddedaim, F. Theoleyre, Implementation of a wsnet module to simulate the IEEE 802.15.4 beacon-enabled mode in multihop topologies, Research Report 00590853 http://hal.archives-ouvertes.fr/hal-00590853, HAL (May 2011).

[4] G. Anastasi, M. Conti, M. Di Francesco, The MAC Unreliability Problem in IEEE 802.15.4 Wireless Sensor Networks, in: International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM), ACM, Tenerife, Spain, 2009, pp. 196–203.

[5] O. Banimelhem, S. Khasawneh, GMCAR: Grid-based multipath with congestion avoidance routing protocol in wireless sensor networks, Ad Hoc Networks 10 (7) (2012) 1346 – 1361.

[6] E. Ben Hamida, G. Chelius, J.-M. Gorce, On the complexity of an accurate and precise performance evaluation of wireless networks using simulations, in: International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM), ACM, Vancouver, Canada, 2008, pp. 395–402.

[7] J. Cartigny, F. Ingelrest, D. Simplot-Ryl, I. Stojmenović, Localized lmst and rng based minimum-energy broadcast protocols in ad hoc networks, Ad Hoc Networks 3 (1) (2005) 1 – 16.

[8] Y. Chen, A. Terzis, On the implications of the log-normal path loss model: an efficient method to deploy and move sensor motes, in: Conference on Embedded Networked Sensor Systems (SenSys), Seattle, USA, 2011, pp. 26–39.

[9] F. Cuomo, E. Cipollone, A. Abbagnale, Performance analysis of IEEE 802.15.4 wireless sensor networks: An insight into the topology formation process, Computer Networks 53 (18) (2009) 3057–3075.

[10] F. Cuomo, S. D. Luna, P. Todorova, T. Suihko, Topology Formation in IEEE 802.15.4: Cluster-Tree Characterization, in: International Conference on Pervasive Computing and Communications (Percom), IEEE, Honk-Kong, 2008, pp. 276–281.

[11] D. S. J. De Couto, D. Aguayo, B. A. Chambers, R. Morris, Performance of multihop wireless networks: shortest path is not enough, SIGCOMM Computer Communication Review 33 (1) (2003) 83–88.

[12] R. de Paz Alberola, D. Pesch, Duty cycle learning algorithm (DCLA) for IEEE 802.15.4 beacon-enabled wireless sensor networks, Ad Hoc Networks 10 (4) (2012) 664—679.

[13] E. W. Dijkstra, Self-stabilizing systems in spite of distributed control, Communications of the ACM 17 (11) (1974) 643–644.

[14] A. Fraboulet, G. Chelius, E. Fleury, Worldsens: Development and Prototyping Tools for Application Specific Wireless Sensors Networks, in: IPSN Track on Sensor Platforms, Tools and Design Methods (SPOTS), Cambridge, USA, 2007.

[15] D.-M. Han, J.-H. Lim, Smart home energy management system using IEEE 802.15.4 and zigbee, IEEE Transactions on Consumer Electronics 56 (3) (2010) 1403–1410.

[16] F. Ingelrest, N. Mitton, D. Simplot-Ryl, A Turnover based Adaptive HELLO Protocol for Mobile Ad Hoc and Sensor Networks, in: International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS), IEEE, Istanbul, Turkey, 2007, pp. 9–14.

[17] N. Karowski, A. C. Viana, A. Wolisz, Optimized asynchronous multi-channel discovery of ieee 802.15.4-based wireless personal area networks, IEEE Transaction on Mobile Computing, to appear.

[18] J. Ko, J. Eriksson, N. Tsiftes, S. Dawson-Haggerty, A. Terzis, A. Dunkels, D. Culler, ContikiRPL and TinyRPL: Happy Together, in: Workshop on Extending the Internet to Low power and Lossy Networks, Chicago, USA, 2011.

[19] A. Koubâa, A. Cunha, M. Alves, E. Tovar, TDBS: a Time Division Beacon Scheduling Mechanism for ZigBee Cluster-Tree Wireless Sensor Networks, Real-Time Systems 40 (3) (2008) 321–354.

[20] P. Levis, T. Clausen, J. Hui, O. Gnawali, J. Ko, The Trickle Algorithm, RFC 6206, IETF (March 2011).

[21] Y. Liu, Q. Zhang, L. Ni, Opportunity-based topology control in wireless sensor networks, IEEE Transactions on Parallel and Distributed Systems 21 (3) (2010) 405–416.

[22] F. Meng, Y. Han, A new association scheme of IEEE 802.15.4 for real-time applications, in: International Conference on Wireless communications, networking and mobile computing (WiCOM), Beijing, China, 2009, pp. 1–5.

[23] P. S. Muthukumaran, R. de Paz, R. Špinar, D. Pesch, MeshMAC: Enabling Mesh Networking over IEEE 802.15.4 through Distributed Beacon Scheduling, in: AdHocNets, Niagara Falls, Ontario, Canada, 2009, pp. 561–575.

[24] B. Nefzi, D. Khan, Y.-Q. Song, TBoPS: A Tree Based Distributed Beacon Only Period Scheduling Mechanism for IEEE 802.15.4, in: International Conference on Distributed Computing in Sensor Systems (DCOSS), Hangzhou, China, 2012, pp. 341–346.

[25] I. Rhee, A. Warrier, J. Min, L. Xu, DRAND: Distributed Randomized TDMA Scheduling for Wireless Ad Hoc Networks, IEEE Transactions on Mobile Computing 8 (10) (2009) 1384–1396.

[26] T. Shu, M. Krunz, S. Liu, Secure Data Collection in Wireless Sensor Networks Using Randomized Dispersive Routes, IEEE Transactions on Mobile Computing 9 (7) (2010) 941 –954.

[27] R. P. Stanley, Acyclic orientations of graphs, Discrete Mathematics 306 (10–11) (2006) 905 – 909.

[28] B. Villaverde, R. De Paz Alberola, S. Rea, D. Pesch, Experimental Evaluation of Beacon Scheduling Mechanisms for Multihop IEEE 802.15.4 Wireless Sensor Networks, in: International Conference on Sensor Technologies and Applications (SENSORCOMM), IARIA, 2010, pp. 226–231.

[29] B. C. Villaverde, S. Rea, D. Pesch, InRout – A QoS aware route selection algorithm for industrial wireless sensor networks, Ad Hoc Networks 10 (3) (2012) 458–478.

[30] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, R. Alexander, RPL: IPv6 Routing Protocol for Low power and Lossy Networks, rfc 6550, IETF (March 2012).

[31] Zigbee, `http://www.zigbee.org/` (2005).